Answers

1 (a) (i) *Systems analysis*: One of the tasks of systems analysis is to observe and document the current business process. This is supplemented by interviewing the user about current procedures as well as future requirements. Documentation such as flowcharts of the current business process should be produced and confirmed with the user. Users are often asked to formally sign-off such documents hence signifying that the analyst has currently understood the current business process. Such diagrams emphasise the flow or sequence of operations in the process and so the sequence of the business process should have been understood if analysis had been carried out correctly.

(ii) *Prototyping:* Prototypes are partially working computer systems used to explore part of a development problem or issue. In many instances it is difficult for the users to understand how part of the system is going to work from looking at paper-based models, screens and reports. In this context it would have been helpful to construct a prototype system of the user interface so that users could experience the 'look and feel' of a significant part of the application before the final system was developed and implemented. Any failure of the system to mirror the sequence of the business process should have been detected at this stage and amendments requested.

(iii) *User acceptance testing*: Systems have to go through a number of testing stages. In user acceptance testing, the potential users of the system run a set of tests to satisfy themselves that the system meets the agreed requirements specification. This formal acceptance of the system is required before a live version can be implemented. The usability of the software is a very important element of user acceptance testing and so the problem of the system not following the business flow should have been picked up at this stage. The system could have been rejected and changes requested.

(b) At this stage there appear to be two clear alternatives to solving the 'business flow' problem.

(1) Change the computer system to reflect the sequence of the business process.

(2) Change the business process to fit in with the screen design.

The first of these options will require programming changes and re-testing of programs and systems. The ease of making these changes depends upon the computer language being used and the availability of tools such as screen and dialogue designers. The alternative option will require changes in business procedures in the housing department and perhaps changes in layout to internal forms. This may be a quicker and easier option to take even if it means changing established processes. However, it may not be possible to change such processes without extensive re-training of staff.

2 (i) *Knowing whether the system is still producing a report or has stopped working*
The Head of Information Technology should ensure that standards are in place to ensure that all software applications display appropriate messages when a process is still in operation (for example, an hourglass or a simple text message such as 'Producing report – please wait'). If the operation takes longer than a few seconds to execute then the software might display a progress indicator in addition to the text message or the hourglass. Progress can be represented by a long rectangular bar that is initially empty but is gradually filled from left to right as the operation proceeds. This can be supplemented with a displayed percentage value of the progress of the task (for example, 70% complete). In all instances the user needs to be sure that an operation is still working so that they do not unnecessarily abort the operation.

(ii) *Speed of report production*
The Head of Information Technology should quantify the speed of report production. As there is conclusive evidence to support the Chief Housing Officer's claims then the software developers should take steps to improve the speed of producing the reports.

These steps might include

Re-structuring the files so that reports need to access fewer files to produce the required data. A common reason for slow report production is that the report program has to navigate across a large number of data files. Hence report production can be speeded up by duplicating data fields, holding cumulative values and derived fields, and other strategies that lead to fewer files being accessed by a particular report.

Providing faster hardware and software.

Defining a separate system dedicated to report production. The slow speed of the current system may be due to the operational and reporting systems running together. Some organisations take a snapshot of the current data and download it to a dedicated system that is used for dedicated report production.

(iii) *The meaning of error messages*

The Head of Information Technology should define standards for error messages and establish standard procedures to ensure that these standards are adhered to. Error messages should be free of computer jargon and contain business terms familiar to the user community. The error message should not only report the error but also suggest a course of action to the user to rectify it.

The error trapping and handling system must be thoroughly tested before the system is implemented to ensure that it works properly.

(iv) *The usability of the user manual.*

The Head of Information Technology should examine the way the user manual has been written and presented. He may wish to consider the following;

Rewriting the manual as a set of smaller handbooks based around functions undertaken by users (for example; raising a Purchase Order, posting a supplier payment etc.). Most users only use parts of a system and so there is no need for them to work their way through a large indiscriminate User Manual.

Asking the users to re-write the User Manual. It might be argued that users are in a better position to write the User Manual because they are more familiar with the business process and the required level of understanding and explanation.

Implementing the on-line HELP facility supposedly promised in the first place and dispensing with the User Manual completely.

Undertaking an analysis to find the most frequent errors. A guide to avoiding and correcting these errors might be produced on a small reference card.

3 (a) FIELD OVERFLOW ERROR IN ORDER TOTAL.

(i) The message probably means that the calculated value (order total) is too large to fit into the field length specified for it. It appears that this message occurs when the order total exceeds £999,999. The field definition probably allows for values up to £999,999.

(ii) The field widths should have been identified in systems analysis. They will be determined from looking at current documents (did purchase invoices exceed £999,999 before the system was developed?), anticipating any increase over the life of the application. The error may not have been found because analysts failed to inspect a sufficiently large sample of purchase orders or failed to confirm the field length with the user. Alternatively, the user may have considered such large orders as unlikely and so confirmed a field width that turned out to be too small. However, this error should have been located in subsequent testing of the boundary values of the system.There is evidence to suggest that many systems fail when the upper and lower boundaries of the data values are entered into the system. Thus the testing should have used the upper limit of order lines (as defined in analysis) with a maximum order quantity allowed (say 99) and a maximum unit price (say 99,999) for each of these lines. These are acceptable (although unrealistic values) and would have triggered the overflow error that was first observed in live operation. The field length of the calculated order total field should reflect the maximum possible field value, not the most realistic.

(b) (i) A software audit trail is a list of transactions created by the software application to record significant user transactions and processes. One author has likened on-line systems data entry to 'writing with invisible ink'. The software audit trail logs information about on-line transactions (such as a user entering purchase order information) so that the transaction can eventually be inspected and verified by internal auditors.The main purpose of the software audit trail is to identify possible fraud. For example, does the value of a supplier invoice match the supplier payment and the original purchase order? It can also identify missing transactions, for example; a supplier payment raised with no matching supplier invoice. Two subsidiary roles of the software audit trail are

– To continually verify the logical and arithmetical correctness of the software.

– To identify user interface problems. For example, do certain transactions have to be frequently corrected by the operator entering the data?

(ii) The software audit trail will probably include the following fields. Four data fields are required in the question.

Transaction-id	A unique transaction number for each record in the audit trail
User-id	The user undertaking the transaction
Date	Date of the transaction
Time	Time of the transaction
Type of transaction	Transaction type (e.g. PO – Purchase Order, SI – Supplier Invoices, SP – Supplier Payment)
Transaction Effect	For example, I – insert, D – delete, A – amend
Prior Value	Value of a transaction field (such as order total) before the process
Post Value	Value of a transaction field (such as order total) after the transaction

(iii) This point could be justified from both points of view. The IT department could argue that a software audit trail is an optional software feature (for example, most spreadsheets do not have one) and so would only be specified and implemented if the user explicitly asks for one. The user departments could argue that a software audit trail is a definite requirement in all finance related systems. They might also make the point that it is difficult for users to specify all non-functional requirements (archiving, back-up, recovery etc. come into a similar category) and so the issue of the audit trail should have been raised by the IT department. On balance, the failure to provide an audit trail was probably the fault of the IT department. A professional approach from the developers should have meant that an audit trail was specified for the system.

4 Three advantages of decentralisation might include;

IT staff become familiar with the objectives and procedures of the operational department. One criticism often voiced of centralised IT departments is that their staff members are divorced from the objectives of the organisation and are unfamiliar with the procedures followed in the operational departments to achieve those objectives. Decentralising staff to the operational departments should address this problem by bringing staff closer to the procedures and systems they are meant to understand and support. In time, IT staff begin to identify with the goals and objectives of the user department (not the IT department or IT community) and their increasing business knowledge means that requirements analysis becomes quicker and better and respect for the IT employees grows within the user department.

IT cost control becomes the responsibility of the head of the operational department. Another common criticism of the centralised approach to IT is that the control of the IT budget of a particular department is not sufficiently under the control of the head of that department. The costs are usually 'cross-charged. from the IT department but the head of the operational department usually finds it difficult to monitor and control those costs and, furthermore, cannot chose to go elsewhere for IT service if he or she feels that the costs are too high. In the decentralised arrangement the IT staff become staff of the operational department and so their activity and their proposals (software and hardware purchase etc.) come directly under the control of the head of the operational department. The head of the department may also choose to spend the budgets elsewhere (outsourcing software development and support) if it is more cost-effective.

Technical solutions are geared to local requirements. It can be argued that centralised IT departments are prone to large strategic visions which result in inappropriate technology being implemented in individual operational departments. Hardware and software solutions may suit the organisation as a whole but they may be too complex for local requirements. Decentralising IT allows department managers to select hardware and software that fits the requirements of a particular department and the skill levels of staff in that department. Many of these solutions are much simpler than the integrated company-wide solution often suggested by the centralised IT department and can usually be implemented quickly and cheaply.

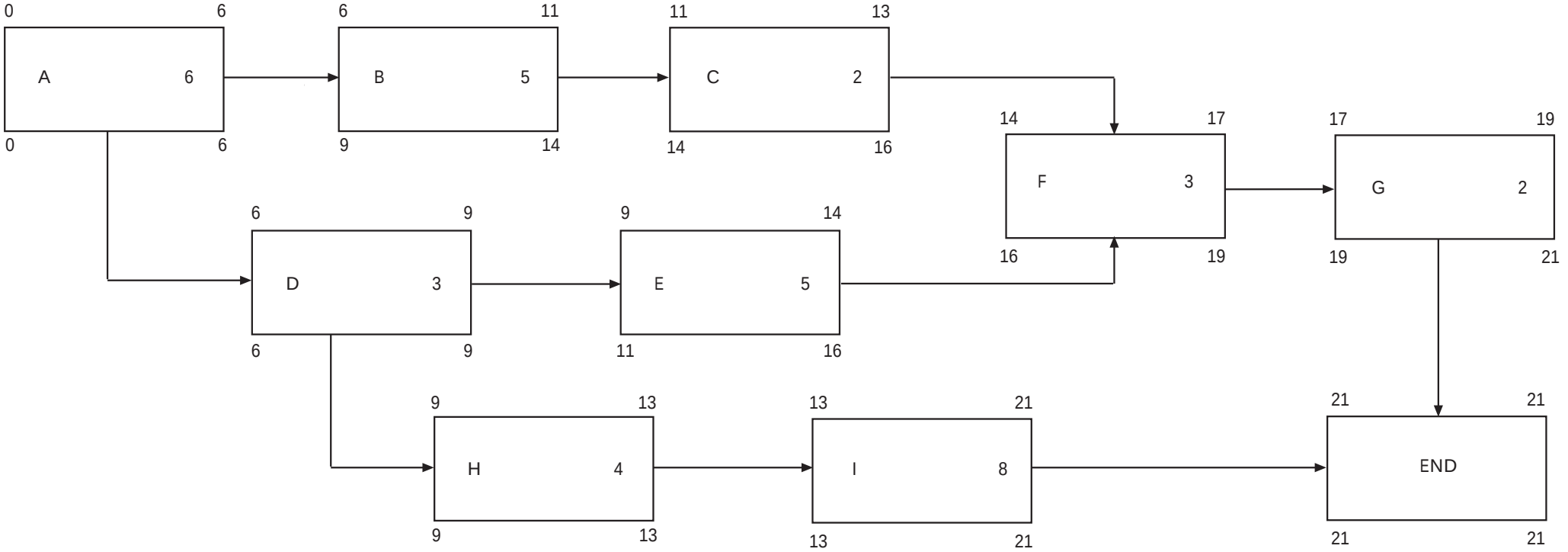
IT staff are on-hand to provide constant support, training and enhancement. In the decentralised arrangement the employees responsible for developing and implementing a software solution are still available in the user department to provide technical support, answer queries and to provide training to new staff. Problems with the system can be quickly identified and implemented without recourse to time consuming change control procedures. The fact that the IT staff have to live with their users provides a powerful motivation to get things right. The option of returning (and hiding) in a protective, centralised IT department has been removed!

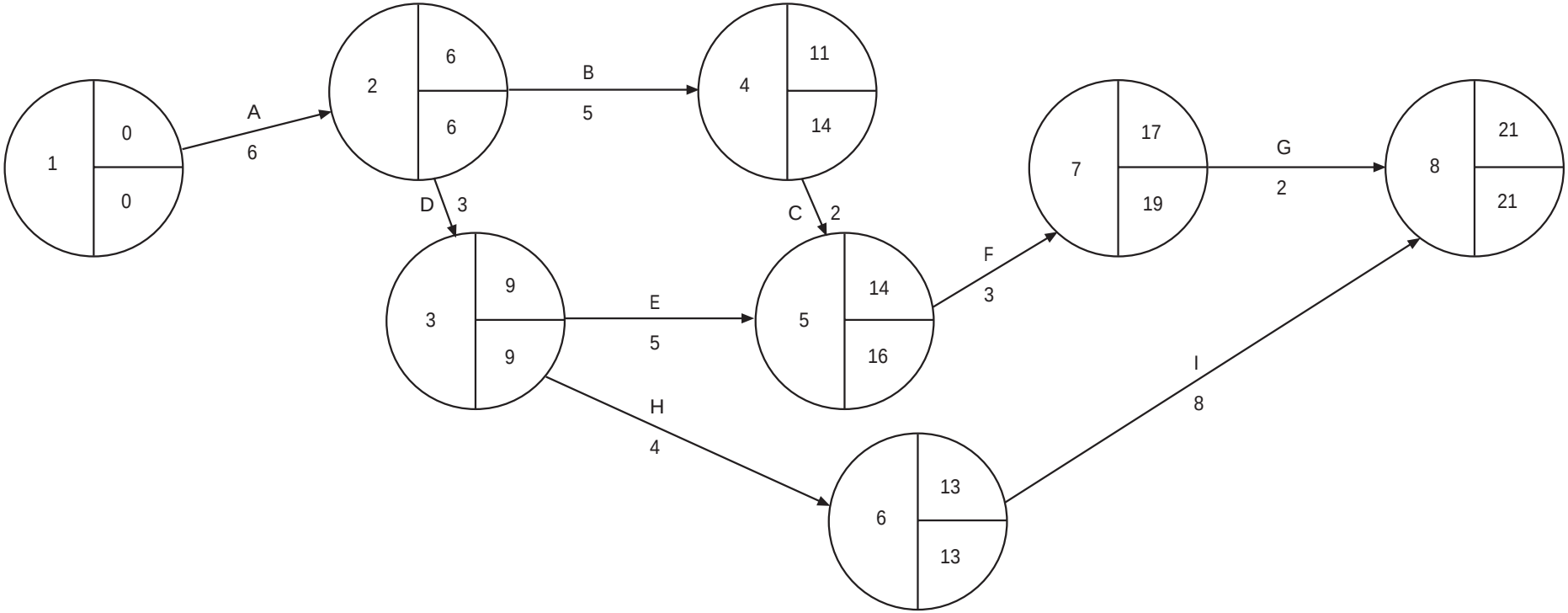
5 **(a)** **(i)** The Critical Path is the sequence of activities that defines the shortest elapsed time of the project. If any of these critical deliverables are delayed then the overall project duration will be extended beyond the minimum time.

(ii) A,D,E,F,G

(iii) Delay in the completion of critical activities will lead to overall project delay. Consequently the project manager should allocate dedicated, experienced resources to those activities with the objective of meeting (or even beating) the original time estimates. The project manager should also monitor more closely the progress of the activities on the critical path. Non-critical activities (such as B) can run over their allotted time (to an extent defined by the difference between the Latest and Earliest Finish Times) without delaying the rest of the project. Hence it is not so important to achieve the original time estimates of these deliverables.

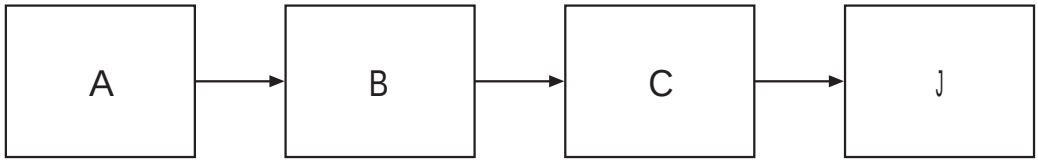
(b) (i) See diagram





(ii) 2 days

(c) No loops are permitted within the project plan because the network essentially shows a series of activities progressing through time. It focuses on the passage of time, not on the successful completion of activities. The fragment is not acceptable because C could not start until B is complete, but B cannot start until C is complete. Consequently B can never start. If the manager feels that the re-work of B is almost certain then a new activity should be defined (say J, rework of B), estimated and inserted into the project sequence (as shown below).



6 **(a)** The entity-relationship model (or logical data model) comprises two main constructs. The rectangular boxes are entities (or more properly entity types). These are significant things or objects that the organisation wants to hold information about. The content of each entity type is shown in the normalised tables at the bottom of the diagram. For example, the organisation has decided that it wants to store information about the entity type ORDER. Each individual occurrence of ORDER is uniquely identified by its order number (order-no). The other attributes of ORDER are shown in the entity type definition.

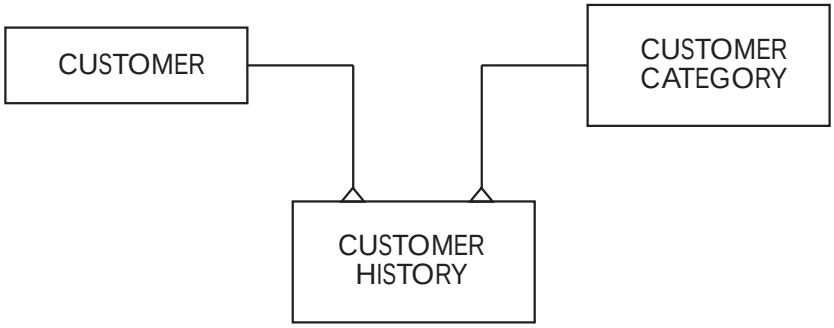
The lines between the boxes represent relationships. They show that business relationships exist between entity types. In this example a CUSTOMER may place many ORDERs, but each ORDER is placed by one CUSTOMER. The many sides of the relationship is shown by a crow's foot symbol.

The relationship is implemented by the key of CUSTOMER (Customer-no) appearing in the ORDER entity type as a foreign key.

(b) The PRODUCT entity type should contain the key of PRODUCT TYPE (product-type-code) as a foreign key to represent the one to many relationship between PRODUCT TYPE and PRODUCT. The error is corrected by adding product-type-code to the PRODUCT entity type and marking it as a foreign key.

The ORDER LINE entity type should not include the field customer-no. ORDER LINE is not directly connected to the entity type CUSTOMER and so this foreign key is not required and should be removed. ORDER LINE is related to ORDER by the foreign key order-no, and ORDER is related to CUSTOMER by the foreign key customer-no, so customer-no is not a foreign key of ORDER LINE.

(c) The model does not currently include attributes for effective date and reason-for-change. One way of integrating these new attributes into the model is to re-define the one to many relationship between CUSTOMER CATEGORY and CUSTOMER as a many to many relationship. A new entity type has to be defined to accommodate this requirement and effective date and reason-for-change are placed in this new entity type. This is shown below. The CUSTOMER HISTORY entity type allows a customer-no to be associated with more than one customer-category-code.



7 **(a)** **(i)** The manager requires an exception report. Exception reports only show data that meets certain criteria. In this example, the exception report would only show products that have gone below their re-order levels or are likely to do so in the next seven days. The criterion for defining the exception is often a variable defined by the user.

(ii) The exception report uses the systems concept of filtering. Filtering mechanisms take 'noise' out of the environment so that the receiver can act upon the required, relevant information. In this example the presence of too much irrelevant information (noise) may lead the manager to make a wrong decision because the noise has distorted the way he interprets (or fails to interpret) the data.

(b) Control systems usually consist of eight elements: input, output, process, feedback, standard, sensor, comparator and effector. Positive feedback usually occurs due to a badly constructed feedback loop where continued inputs lead to deterioration in the system performance rather than its improvement. The problem causing the positive feedback might be anywhere in the feedback loop (for example, an incorrect standard or an inadequate sensor). Whatever the reason, systems with positive feedback increasingly get out of control.

Negative feedback occurs in systems with correctly constructed feedback loops where deviations from the required standard are properly identified. Changes are then made to the inputs of the system to bring the performance of the system back to its required standard. Systems with negative feedback will perform to the criteria specified in the system's standard. The system is being correctly controlled.

(c) The long established family business is an example of a relatively closed (or semi-closed) system. Such a system may have relatively little interaction with its environment, depending on a business strategy developed by a Board of Directors taken from a fairly homogenous group (the family). Such systems tend to experience an increase in entropy, characterised by increased uncertainty and disorganisation. In business terms this often leads to business failure. Entropy may be countered by bringing in new information from outside the usual environment. The Information Systems Director has provided this 'negative entropy'. It is likely that the 'breath of fresh air' refers to new ideas and inputs which have made the organisation more 'open', reflecting this better interaction with its environment in its improved business performance.

8 **(a)** **(i)** This particular type of malicious program is usually called a bomb or time-bomb.

(ii) This type of program lies dormant in the system until it is activated by some event or combination of events. In this instance that event appears to be time (date = April 1) and perhaps use by the Personnel department (Department = Personnel). This would appear as an IF statement in the program

```
IF date = April 1 AND Department = Personnel
Display abusive message
ENDIF
```

(iii) The organisation will have to organise a series of structured walkthroughs of the program code written by the redundant programmer. Inspecting the code for future logic and time bombs is the most effective way of finding further instances.

(b) **(i)** The programmer will have committed the unauthorised modification offence, in that he has caused an unauthorised modification of the contents of a computer system. This is a crime under the U.K. Computer Misuse Act.

(ii) Hacking is concerned with gaining unauthorised access to a computer system. It is usually associated with people who are not employees of the organisation who wish to access an organisation's data for mischievous or malicious intent. However, in its widest term – a person who gains access to a computer without permission – it can also be applied to company employees themselves. Hacking has been made possible by organisations using open telecommunications networks which are also accessible to the hacker via powerful workstations and modems.

(c) **(i)** A computer virus is a computer program written to cause damage or disruption to a computer system. A programmer usually writes the virus for mischievous rather than malicious reasons. The virus is usually unwittingly distributed on floppy disks and CDs or over a network. Once the virus comes into contact with a system it replicates itself onto the system and lies dormant until either the use of the system, some defined event or transaction or a certain date activates it. The replication of the virus makes it almost impossible to find its original source.

(ii) Counter measures include:
Purchasing anti-virus software and regularly updating it with new releases. This anti-virus software detects known viruses and destroys them. This virus may be known and identifiable by the anti-virus software. However, it must be recognised that such software only protects against known viruses.

Implementing stringent internal procedures to make sure that unauthorised disks and CDs are not used within the organisation. Most organisations require their employees to have external disks and CDs checked before they can be used internally. Failure to comply with these requirements usually leads to disciplinary action, including dismissal.

The organisation also needs to implement external checking procedures to protect it from unauthorised access. This will include physical security, firewalls and software passwords.

The organisation may decide that user workstations do not have external disk or CD drives.

9 (a) (i) Operations

The computer operations staff are responsible for the day-to-day running of computer equipment and computer systems. They are responsible for ensuring that jobs within the company are executed to their defined schedule. For example, the payroll program may have to be run every Friday at 14.00 hours. They may also be responsible for ensuring that all inputs for a defined program have been received (for example, all time sheets for payroll processing) as well as making sure that all outputs (payslips) are properly produced and distributed. Operations will also monitor machine and system use and respond to hardware problems and faults. They also make sure that the correct disks are loaded for scheduled jobs and that printers are working and stocked with paper.

(ii) Database Administration

The Database Administration is usually concerned with setting up and maintaining physical data files and databases. In some instances they are also responsible for logical data design (entity-relationship models) as well as defining the relationship between the organisation's logical and physical data model. Database administration will also be concerned with practical issues of database use such as data integrity and accuracy, access rights of users and backup, security and recovery procedures. In many instances Database Administration is also responsible for defining and maintaining the logical and physical data dictionary.

(iii) Standards

The Standards section will be responsible for defining and maintaining standards throughout the Information Systems department. In the first instance this section will define (or adopt) standards for defining the quality and completeness of all work within the section. These standards will extend from feasibility (so, for example, there may be a standard method of making the business case for an application) through to program coding and testing. Once these standards have been defined and agreed, the standards section will be involved in ensuring that they are adhered to. This will normally involve attendance at structured walkthroughs where they will help assure the quality of products and ensure that they are fit for purpose and release.

(b) (i) In a 'flat, project-based structure', project teams are created to undertake specific projects. These teams exist as long as the project exists. Once the project finishes, then the team is disbanded and the members are re-allocated to another team. The team itself has no place in a hierarchy. Indeed the team itself will vary in size and composition as the project progresses through the systems development lifecycle. Normally, such teams are constructed from team members on wide salary bands in a department where there may only be three or four bands. Job titles are very generic and there are usually a large number of people on each grade. Within each project team, employees will take on roles that may differ from project to project.

(ii) Advantages

Computer systems development is essentially a project-based activity. The number of employees required for each project varies with the number of projects being commissioned, as well as the stage reached in each project. For example, more employees can be used at the programming stage than at the Feasibility Stage because more activities can be done in parallel. This need for flexibility is supported by the 'flat, project-based structure' where employees can be drafted in as needed. This flexibility is supported by the concept of wide salary bands where technical expertise can be rewarded almost as much as management expertise. These wide bands also help de-couple the relationship between salary and job. In the project-based flat structure a project manager on one project may become a programmer on the next. Thus employees adopt roles in projects that are not tied to some position in a management hierarchy. This allows organisations to flexibly staff projects without incurring problems of status and job grading and having one part of the workforce over-stretched whilst some other part is underworked!

Certificate Examination – Paper 5
Information Analysis **Marking Scheme**

1 **(a)** One mark for each valid point up to a maximum of four marks for each stage. There are three stages giving a total of 12 marks.

(b) Up to two marks for each suggestion. Two suggestions required giving a total of 4 marks.

2 One mark for each valid point up to three marks for each problem. Four problems required giving a total of 12 marks.

3 **(a)** **(i)** One mark for the meaning of the error message. Up to two marks for the reason why it occurred.

(ii) Two marks for each valid point up to a maximum of 4 marks.

(b) **(i)** One mark for each valid point up to a maximum of four marks.

(ii) One mark for each valid field. Four fields required giving four marks.

(iii) One mark for each valid point up to a maximum of three marks.

4 One mark for each valid point up to a maximum of three marks. Three advantages required giving a total of nine marks.

5 **(a)** **(i)** One mark for each valid point up to a maximum of two marks.

(ii) One mark for the correct activities.

(iii) One mark for each valid point up to a maximum of two marks.

(b) **(i)** One mark for correct insertion of H and I.
One mark for correct (new) project duration on final node. Project duration does not have to be stated explicitly – it's sufficient to show it on the diagram.
Up to two marks for correct backward pass values on the old network activities.

(ii) Two marks for the correct answer.

(c) Up to two marks for explaining why the suggestion is unacceptable. Up to two marks for the solution.

6 **(a)** Up to two marks for the explanation of entities. Up to three marks for the explanation of relationships.

(b) One mark for each error and one mark for the correction. Two errors are specified.

(c) Two marks for each valid point up to a maximum of six marks.

7 **(a)** **(i)** Identification of the type (1 mark) and brief explanation of its meaning (1 mark).

(ii) Identification of filtering (1 mark) and brief explanation of its meaning (up to 2 marks).

(b) One mark for each valid point up to a maximum of five marks.

(c) Up to three marks for open/closed system issues.
Up to two marks for entropy issues.

8 **(a)** **(i)** One mark for bomb.

(ii) One mark for each valid point up to a maximum of two marks.

(iii) One mark for each valid point up to a maximum of two marks.

(b) **(i)** One mark for the specific legislation.
Up to two marks for the specific offence.

(ii) One mark for each valid point up to a maximum of two marks.

(c) **(i)** One mark for each valid point up to a maximum of two marks.

(ii) One mark for each valid point up to a maximum of three marks.

9 **(a)** **(i)** One mark for each valid point up to a maximum of three marks.

(ii) One mark for each valid point up to a maximum of three marks.

(iii) One mark for each valid point up to a maximum of three marks.

(b) **(i)** One mark for each valid point up to a maximum of three marks.

(ii) One mark for each valid point up to a maximum of three marks.